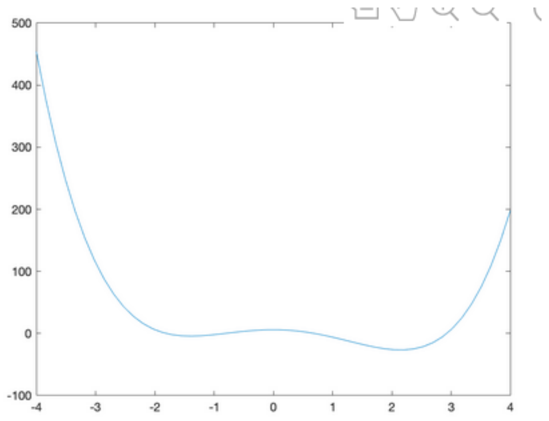
Homework 4 Solution

CS6923 Machine Learning

Part I

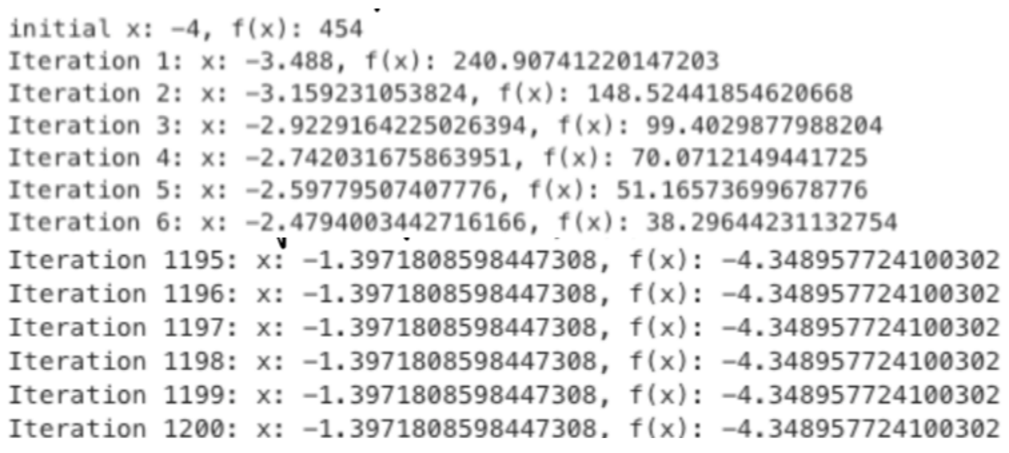
1. (a) **(1 point)**



Graph above is f(x) in range [-4, 4].

First, get the derivation of f(x): $f'(x) = 8x^3 - 6x^2 - 24x$
Set f'(x) = 0, we get the roots:
$x_1 = 0$, $x_2 = 2.147$, $x_3 = -1.397$
Since 8 > 0, f(x) decrease in range(-4, -1.397) and (0, 2.147)
increase in range(-1.397, 0) and (2,147, 4)
We get $f(x_2) = -26.612$ and $f(x_3) = -4.349$
So, the value of x at the local minimum is -1.397
at the global minimum is 2.147

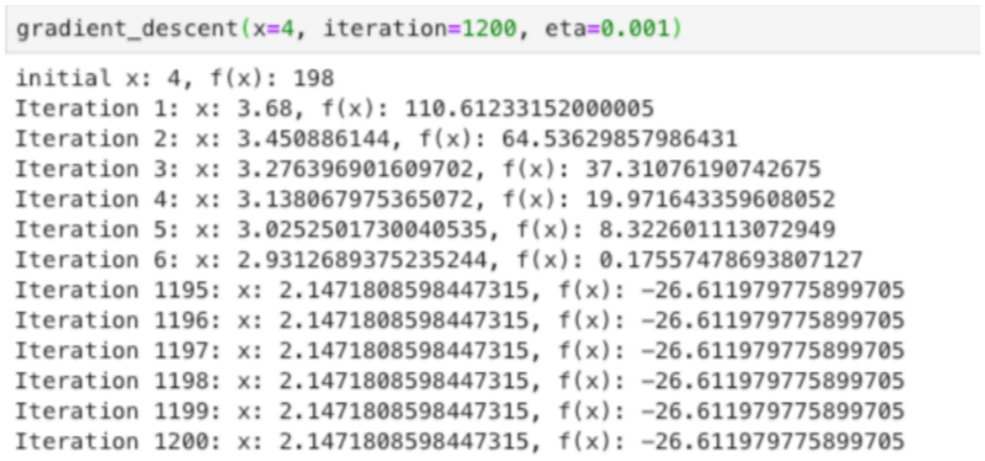
(b) **(0.5 point)**



```
initial x: -4, f(x): 454
Iteration 1: x: -3.488, f(x): 240.90741220147203
Iteration 2: x: -3.159231053824, f(x): 148.52441854620668
Iteration 3: x: -2.9229164225026394, f(x): 99.4029877988204
Iteration 4: x: -2.742031675863951, f(x): 70.0712149441725
Iteration 5: x: -2.59779507407776, f(x): 51.16573699678776
Iteration 6: x: -2.4794003442716166, f(x): 38.29644231132754
Iteration 1195: x: -1.3971808598447308, f(x): -4.348957724100302
Iteration 1196: x: -1.3971808598447308, f(x): -4.348957724100302
Iteration 1197: x: -1.3971808598447308, f(x): -4.348957724100302
Iteration 1198: x: -1.3971808598447308, f(x): -4.348957724100302
Iteration 1199: x: -1.3971808598447308, f(x): -4.348957724100302
Iteration 1200: x: -1.3971808598447308. f(x): -4.348957724100302
```

The value of x has converged. It found the local minimum.

(c) **(0.5 point)**

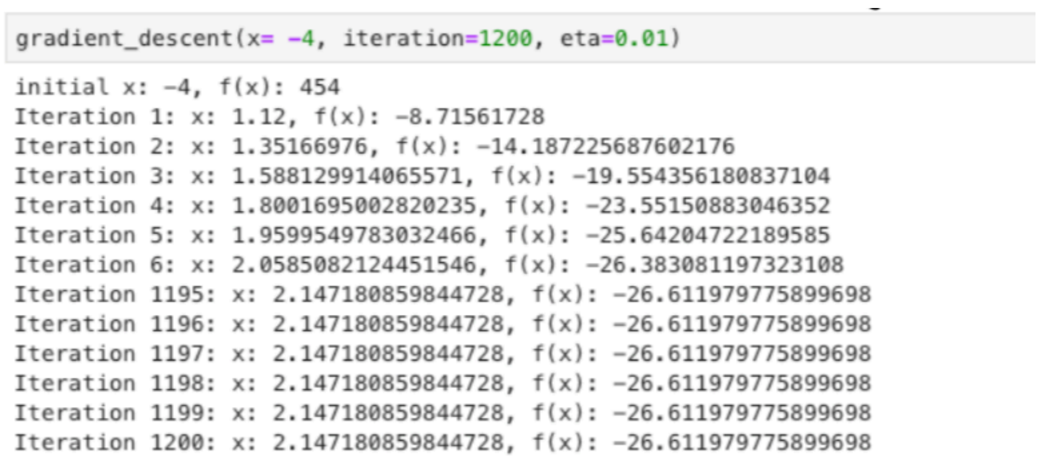


```
gradient_descent(x=4, iteration=1200, eta=0.001)

initial x: 4, f(x): 198
Iteration 1: x: 3.68, f(x): 110.61233152000005
Iteration 2: x: 3.450886144, f(x): 64.53629857986431
Iteration 3: x: 3.276396901609702, f(x): 37.31076190742675
Iteration 4: x: 3.138067975365072, f(x): 19.971643359608052
Iteration 5: x: 3.0252501730040535, f(x): 8.322601113072949
Iteration 6: x: 2.9312689375235244, f(x): 0.17557478693807127
Iteration 1195: x: 2.1471808598447315, f(x): -26.611979775899705
Iteration 1196: x: 2.1471808598447315, f(x): -26.611979775899705
Iteration 1197: x: 2.1471808598447315, f(x): -26.611979775899705
Iteration 1198: x: 2.1471808598447315, f(x): -26.611979775899705
Iteration 1199: x: 2.1471808598447315, f(x): -26.611979775899705
Iteration 1200: x: 2.1471808598447315, f(x): -26.611979775899705
```

The value of x has converged. It found the global minimum.

(d) **(0.5 point)**



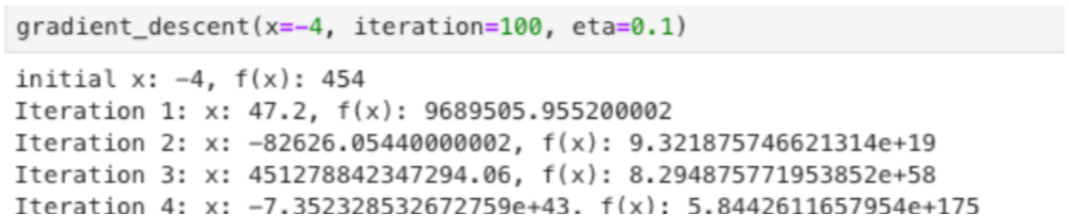
```
gradient_descent(x= -4, iteration=1200, eta=0.01)

initial x: -4, f(x): 454
Iteration 1: x: 1.12, f(x): -8.71561728
Iteration 2: x: 1.35166976, f(x): -14.187225687602176
Iteration 3: x: 1.588129914065571, f(x): -19.554356180837104
Iteration 4: x: 1.8001695002820235, f(x): -23.55150883046352
Iteration 5: x: 1.9599549783032466, f(x): -25.64204722189585
Iteration 6: x: 2.0585082124451546, f(x): -26.383081197323108
Iteration 1195: x: 2.147180859844728, f(x): -26.611979775899698
Iteration 1196: x: 2.147180859844728, f(x): -26.611979775899698
Iteration 1197: x: 2.147180859844728, f(x): -26.611979775899698
Iteration 1198: x: 2.147180859844728, f(x): -26.611979775899698
Iteration 1199: x: 2.147180859844728, f(x): -26.611979775899698
Iteration 1200: x: 2.147180859844728, f(x): -26.611979775899698
```

The value of x has converged. It found the global minimum.

Though it has the same start with (b), with different value of eta, it converges to different local minimum.

(e) **(0.5 point)**



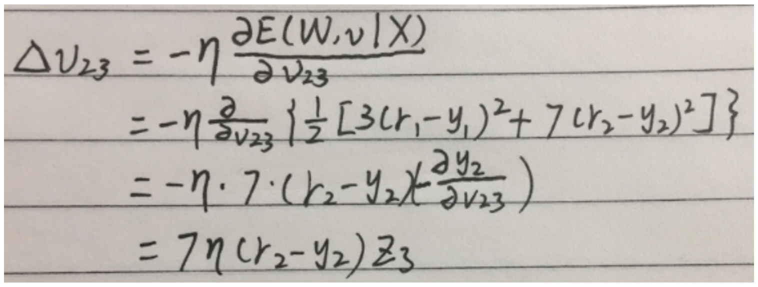
```
gradient_descent(x=-4, iteration=100, eta=0.1)

initial x: -4, f(x): 454
Iteration 1: x: 47.2, f(x): 9689505.955200002
Iteration 2: x: -82626.05440000002, f(x): 9.321875746621314e+19
Iteration 3: x: 451278842347294.06, f(x): 8.294875771953852e+58
Iteration 4: x: -7.352328532672759e+43, f(x): 5.8442611657954e+175
```

The value of x and f(x) overflow after iteration 4.

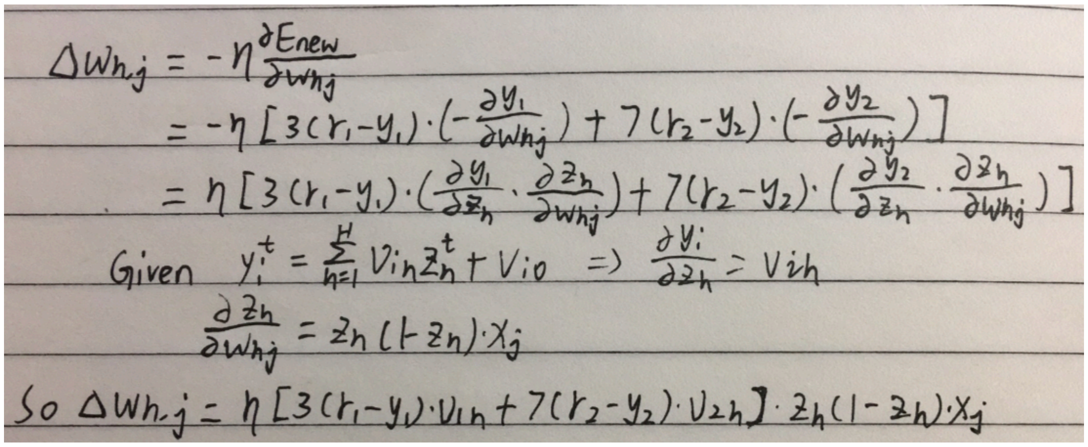
Thanks to Mengwei Ren and Daitao Xing for the above screenshots.

2. (a) **(0.5 point)**
It updates 500 * 100 = 50000 times.
(b)(i) **(1 point)**



$$\Delta v_{23} = -\eta \frac{\partial E(W,v|X)}{\partial v_{23}}$$
$$= -\eta \frac{\partial}{\partial v_{23}} \left\{ \frac{1}{2} \left[3(r_1-y_1)^2 + 7(r_2-y_2)^2 \right] \right\}$$
$$= -\eta \cdot 7 \cdot (r_2-y_2)\left(-\frac{\partial y_2}{\partial v_{23}}\right)$$
$$= 7\eta (r_2-y_2) z_3$$

(b)(ii) **(1 point)**



$$\Delta w_{hj} = -\eta \frac{\partial E_{new}}{\partial w_{hj}}$$
$$= -\eta \left[3(r_1-y_1)\cdot\left(-\frac{\partial y_1}{\partial w_{hj}}\right) + 7(r_2-y_2)\cdot\left(-\frac{\partial y_2}{\partial w_{hj}}\right)\right]$$
$$= \eta \left[3(r_1-y_1)\cdot\left(\frac{\partial y_1}{\partial z_h}\cdot\frac{\partial z_h}{\partial w_{hj}}\right) + 7(r_2-y_2)\cdot\left(\frac{\partial y_2}{\partial z_h}\cdot\frac{\partial z_h}{\partial w_{hj}}\right)\right]$$
Given $y_i^t = \sum_{h=1}^{H} v_{ih} z_h^t + v_{i0} \Rightarrow \frac{\partial y_i}{\partial z_h} = v_{ih}$
$$\frac{\partial z_h}{\partial w_{hj}} = z_h(1-z_h)\cdot x_j$$
So $\Delta w_{h,j} = \eta\left[3(r_1-y_1)\cdot v_{1h} + 7(r_2-y_2)\cdot v_{2h}\right]\cdot z_h(1-z_h)\cdot x_j$

3. (a) **(0.5 point)**
None
(b) **(0.5 point)**
NeuralNetCK
(c) **(0.5 point)**
NeuralNetCK
It has multiple outputs and sum up to 1.
(d) **(0.5 point)**
NeuralNetZeroOne
It has multiple outputs and they are independent.

4. (a) **(1 point)**
It's not a good idea to do this for a neural net, because the neural net is using the attributes as numerical input values and computing weighted sums of the input values. The 1,2,3,4 encoding means that e.g., almond and anise (1 and 2) are closer together numerically than almond and fishy (1 and 4), making it harder to distinguish between almond vs fishy, than between almond vs. anise. There is no reason that this should be the case, and it can make it harder to train a neural net to do well on the problem.

In the case of random forests: if the random forest changes the values to 1,2,3,4, but continues to treat the attribute as a categorical attribute, then the change makes no difference. If the attribute is placed in a node of a tree, then it will have 4 children corresponding to the 4 values of the attribute.

If, however, the random forest is treating the attribute as a numerical value, then it's generally not a good idea to encode it as 1,2,3,4. The reasons are similar to the reasons for neural nets.

(b)(i) **(1 point)**

	Z1	**Z2**	**Z3**	**Z4**	**Z5**	**Z6**	**Label**
$X^{(1)}$	0	0	0	1	1	0	0
$X^{(2)}$	0	0	1	0	0	1	0

(b)(ii) **(1 point)**

If the values are low, medium, high, then there is an ordering: low < medium < high. Low is closer to medium than it is to high. One hot encoding treats low, medium, and high as three arbitrary values, and you lose the information about the relation between low, medium, and high. Using 1,2,3 preserves that relation and learning algorithms can take advantage of that.

That said, the encoding 1,2,3 treats the values as being equally spaced, and this may not be appropriate.

(b)(iii) **(1 point)**

When there are only two values, we don't have the problems discussed above. First, we don't have the problem that one pair of values ends up being closer together than another pair of values.

Even if the two values are low and high, and we set them to 1 and 0 (so that value(low) > value(high)) this is unlikely to be a problem. For example, in neural nets, we would be multiplying the value by a weight, and since the weight can be negative, this would effectively reverse the ordering (e.g.with the weight -1, we have -1*value(low) > -1*value(high)).

Part II

(a)**(1 point)**

'.' ',' 'the' 'a' 'and'

(b) **(1 point)**

'bad' 'best' 'n't' 'too' 'moving'

(c) **(1 point)**

Any reasonable answers.

(d) **(1 point)**

Accuracy: 54.6%

(e) **(1 point)**

If you increase the number of attributes, you're adding more information about the examples, and the learning algorithm might be able to use that information to achieve higher accuracy.

However, the additional attributes might by chance have reasonably high information gain on the examples in the training set. If the learned hypothesis incorporates these attributes, this can lead to incorrect predictions. A related problem is that the use of additional attributes may lead to more complex hypotheses, and there is a danger of overfitting.

(f) **(1 point)**

Any reasonable answers.